

Edith Cowan University

Research Online

Australian eHealth Informatics and Security
Conference

Conferences, Symposia and Campus Events

12-4-2013

Developing a Mobile Audiometric Sound Booth Application for Apple IOS Devices

Luke J. Brook

Edith Cowan University, l.brook@ecu.edu.au

Patricia A.H. Williams

Edith Cowan University, trish.williams@ecu.edu.au

Follow this and additional works at: <https://ro.ecu.edu.au/aeis>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Brook, L. J., & Williams, P. A. (2013). Developing a Mobile Audiometric Sound Booth Application for Apple IOS Devices. DOI: <https://doi.org/10.4225/75/57980c9531b3d>

DOI: [10.4225/75/57980c9531b3d](https://doi.org/10.4225/75/57980c9531b3d)

2nd Australian eHealth Informatics and Security Conference, held on the 2nd-4th December, 2013 at Edith Cowan University, Perth, Western Australia

This Conference Proceeding is posted at Research Online.

<https://ro.ecu.edu.au/aeis/10>

DEVELOPING A MOBILE AUDIOMETRIC SOUND BOOTH APPLICATION FOR APPLE IOS DEVICES

Luke J Brook¹, Patricia A H Williams²

¹School of Computer and Security Science,

²eHealth Research Group, School of Computer and Security Science, Security Research Institute
Edith Cowan University, Perth, Australia

¹l.brook@ecu.edu.au, ²trish.williams@ecu.edu.au

Abstract

Hearing loss in Australian children, particularly those in rural and remote areas, is a growing health concern. Resultant deafness has significant impact on the educational and social development of such children. Whilst telehealth has provided other benefits for rural and remote areas, the absence of suitable testing equipment and associated healthcare specialists has meant that there is a gap in this area of child health that could be addressed using a mobile solution. This paper discusses the research and development of a mobile application for testing and diagnosis of hearing loss in children. It is unique in its ability to be able to function in uncontrolled test environments, and conforms to current hearing assessment standards. The application also produces an audiogram that can be immediately electronically transferred to the relevant healthcare provider. The application is specifically designed for young children to use and is designed to be used in collaboration with healthcare specialist audiologists and paediatricians. The prototype is currently undergoing testing at the Telethon Speech and Hearing institute, Perth, Western Australia. The results are expected to deliver crucial quantitative data to determine the accuracy of the application, as well as qualitative data from healthcare professionals and consumers.

Keywords

Ehealth, Mobile Applications, Healthcare Software, Hearing Loss, Health Informatics.

INTRODUCTION

Hearing loss in Australian indigenous children is becoming an increasing concern for the public health sector, with the percentage of affected children comparable to that of third world countries (Weber, 2012). In metropolitan areas the number of aboriginal children who suffer from some form of hearing loss is as high as 75%, while in regional communities this is up to 94% (O'Keefe, 2012). This is commonly attributed to untreated ear infections called Otitis Media (OM), a general term used to describe inflammation of the middle ear (Leach, 1999). The problem needs to be addressed to ensure a higher quality of life for the children affected.

Over the past few years there has been significant progress made with telehealth solutions, including the recent study into *The feasibility of a community-based mobile telehealth screening service for Aboriginal and Torres Strait Islander children in Australia* (Bensink et al., 2010). The study discovered an alarming number of children who were unaware that they suffered from hearing loss, and had not been diagnosed due to lack of immediate and regular access to health care professionals to seek diagnosis and treatment of ear infections (Bensink et al., 2010).

Without additional solutions to address the issue and aid in the diagnosis of hearing loss in children, the impact in the long term is potentially dire. A child's growth and development occurs rapidly throughout early childhood, and thought to be heavily influenced by their environment and experiences (Stevens, 2013). Issues such as undiagnosed hearing loss can have an ongoing impact throughout life, and the inability to hear, or hear well, is proven to cause delay in receptive and expressive communication skills, can cause social isolation and poor self-concept, and result in poor performance in classrooms, resulting in the inability to learn (O'Keefe, 2012; Timms, Grauaug, & Williams, 2012).

To address this issue a software application has been developed, as part of a research study, for testing and diagnosis of children's hearing using a mobile device in an uncontrolled testing environment. The platform

chosen for development was Apple's iOS for smart phone and tablet computer devices and the project was facilitated with collaboration with speech and hearing healthcare professionals. The mobile application may be used to complement the other solutions, in a holistic approach, toward the diagnosis of hearing loss. For instance it could be used in conjunction with the Remotoscope lens, developed by Georgia Tech and Emory University (Korschun, 2012), which uses a device to fit over an iPhone camera to take pictures of the ear canal and ear drum. The resultant images can then be sent via email to specialists, such as audiologists and paediatricians. By combining the two technologies, specialists would ideally be able to determine if a child is suffering from hearing loss, whether it is attributed to an infection, and if a hearing loss persists once an infection has been treated. This paper details the structure and development process for this application.

THE DEVELOPMENT PROCESS

The primary concern when developing such an application was ensuring the validity and reliability of future research, and ensuring results can be reproduced. In order to achieve this, it was important early in the design process to select a platform where all apparatus were configured in the same manner. Although other mobile and tablet device operating system vendors specify minimum hardware requirements, to run the operating systems, if chosen as the target platform for the project, the lack of standardised hardware design may have led to issues for the proposed research. Apple's iOS operating system was selected as the platform for the project, which includes Apple's recent generations of smart phone and tablet devices, such as the iPhone and iPad. The reason for this selection of devices is due to Apple's limitation of the availability of their operating system to first party devices, with standardised hardware specifications for each generation of devices ("What is iOS," 2013).

This standardisation of hardware is not a feature shared amongst Apple's competitors, including Google's open-source Android operating system ("Porting Android to Devices," n.d.) and Microsoft's Windows Phone 7/8 ("Windows Phone- Phones," n.d.), which allow third party hardware manufacturers to bundle their operating systems in their devices. For the development of iOS applications for Apple devices, Apple encourages developers to utilise their software development kit (SDK), the Xcode development tools package. Additionally, all developers are required to register for the Apple iOS Developer program, allowing deployment and testing of software builds on physical devices ("iOS Developer Program ", 2013). The Xcode developer tools package is free to download, and includes an integrated development environment (IDE), with Interface Builder design tool and Apple low-level virtual machine (LLVM) compiler, and an iOS simulator for testing and debugging ("Develop," n.d.). It is important to note that the Xcode developer tools package is exclusive to the Apple computer operating system.

For the development of this project, only certain assets were utilised from the Apple developer portal, as the actual development of the application was performed using additional tools such as Adobe Flash Pro CS5.5 with Adobe AIR. The following defines the different resources utilised during the application development process.

RESOURCES

As the project was a learning and development task, a significant amount of information was sourced from Apple's Developer Program library ("iOS Dev Center," 2013) and Adobe's Support and Learning library ("Adobe Help and Support," 2013). The application was also developed with the following resources:

- 2.3GHz Intel Core i5 13" MacBook Pro running OS X 10.7.5
- 2.6 Ghz Intel Core i5 Desktop PC running Microsoft Windows 8 Pro
- Apple iPod 8GB 4th Generation
- Apple iPhone 5, 32GB
- Apple iPad 4 64GB
- Adobe Flash Professional CS5.5/6, utilising ActionScript 3 with Adobe AIR
- Apple iOS 6 SDK with Xcode 4.6
- Audacity
- Third octave band warble tone sound files

The tones have been developed by Dr Stephane Pigeon, PhD engineer and professional sound engineer, and differ from other tones, as they do not require specific headphones, minimise room and headphone resonance. The tones are based on the international standard ISO389-7:2005, which is recommended by the British Society of Audiology (Pigeon, 2013).

TESTING AND DEBUGGING

Although a majority of the application development was completed on a Microsoft Windows based PC, all testing and a majority of debugging was performed on an Apple MacBook Pro. Flash Professional CS5.5 does include a graphical debugger on Microsoft Windows, but advanced debugging and simulation features require Apple's Xcode SDK, which is restricted solely to Apple computers. While the Xcode simulator is suitable for simulating high-level APIs, such as the touch input layer interface, it is not an emulator and cannot be used for performance testing and low level APIs. In order to conduct thorough testing, physical iOS devices were required, and an Apple MacBook Pro was used for the creation of certificates and provisioning profiles, required for compilation and signing of the app package file, as well as deployment of the app onto the iPod and iPhone devices.

Comparison of the test tones on the device was performed by using a male-male cable with 3.5mm headphone jacks to connect iOS devices to the microphone port of a computer running Audacity. Tones were then procedurally played from the device, and recorded in Audacity. A comparison of the input tone and the original tones was then conducted to verify the tones played on the device were identical.

HARDWARE AND SOFTWARE CONSIDERATIONS

The hearing test application has been designed to test children's hearing, and store results on the device's local storage, allowing results to be shared with health professionals via email. Unfortunately, as all apps need to be compiled into a single package and signed prior to deployment, any files stored within the package are not able to be edited. There are also restrictions on where newly created files can be stored and limitations on which hardware features may be accessed.

The exception to this is the allocated self-contained storage available to applications. Originally, the proposed solution was to utilise this storage space, although it was later discovered that iOS restrictions prevent third party applications from accessing other application storage. The possibility of creating a file storage system and linkage to the iOS mail client from within the application was reviewed, but ultimately it was decided that such a solution was outside the scope of the development project. The final solution to the problem was to utilise the iOS device's screen capture feature which is normally achieved by pressing the device's *Home* and *Power* buttons simultaneously. The feature has since been hard-coded and assigned to a save button on the Audiogram frame. Screen captures are then saved to the device's camera roll, where they can be easily accessed outside of the application for review, and forwarded to health professionals using the iOS device's inbuilt email application and messaging features.

THE APPLICATION

Technical design

The project produced an application potentially capable of emulating an audiometric sound booth, utilising either passive noise cancelling headphones or bone vibrators for testing conductive hearing mechanisms. The final application consists of four components, and the process is demonstrated in Figure 1:

- Main Menu, with preceding menu to enter the name of the participant;
- Background noise assessment tool;
- Interactive story and posting box audiometric sound booth emulation tool;
- A final audiogram graph of results.

The background noise assessment tool is used to sample background audio over a period of time, providing the user with a final reading of the average background noise. These results are then used to advise the user whether their environment is suitable to perform the test.

The posting box component flows in a manner that advances the player through several degrees of frequency, with up to eighteen levels of volume per frequency, although the number of sounds played-per-frequency is dependent on the child's hearing ability, determined by a divide and conquer algorithm.

Results of the test are then collated, and presented in an audiogram at the end of the test, which displays the date, time, user's name and summary of the background noise analysis, with the option to save the audiogram to the device's local storage.

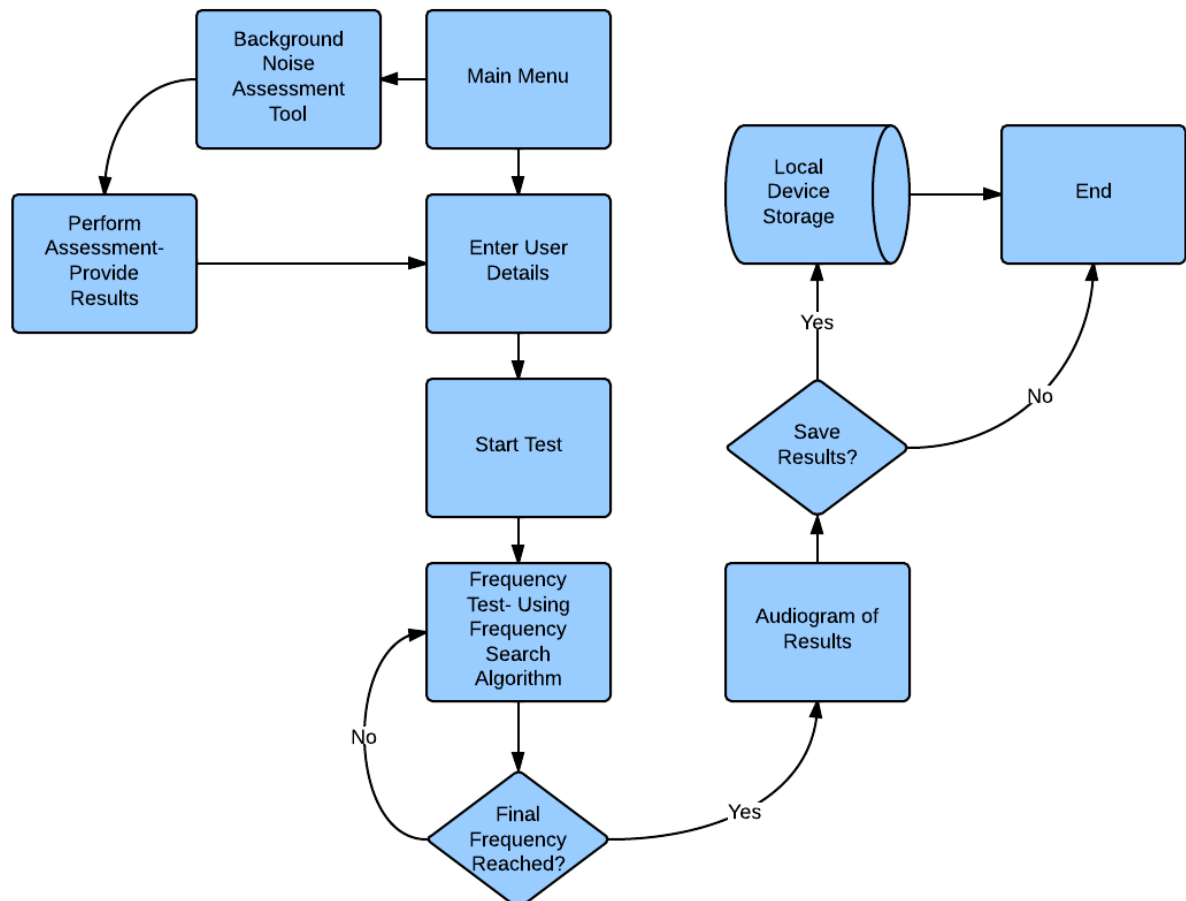


Figure 1 – Application Structure Map

APPLICATION DESIGN AND FUNCTIONALITY

The application has been designed with a display resolution of 320x480px, previously the standard for Apple iPod and iPhone development, with the exception of the newer iPhone 5/5s devices. All elements of the application have been developed using scalable components and vector graphics to allow for scaling on larger devices.

As the application takes the form of an interactive posting-box game, the primary form of interaction is through touch, where users are required to drag a token into a container when they hear a tone (Figure 2). The touch and drag design was an important design choice, due to the ages of the children the application is intended for, and this is a universally accepted children's tactile activity. These methods of interaction between the user and the application are native to Apple iOS devices, which include touch-down, touch-release and touch-drag methods of interaction through finger pointers and swipe actions. Other user input, such as the soft keyboard, are implemented through the Apple iOS's existing features, which also rely on touch input.

Additionally, the touch-drag method of interaction was selected to emulate typical audiometric posting box play-testing, employed by audiologists when performing a hearing screening for children, where children hold an

object to their ear, such as a coloured token, shell or animal, and place the object into a container when a sound is heard.

To further encourage child participation, and to put the posting box game into context, the game component has been integrated with an interactive story based on Eric Carle's *The Very Hungry Caterpillar*. Children are introduced to the test with a narrated story, explaining the caterpillar is hungry, and to listen for the sound before feeding him an item of food. If the child's response falls within the given response time, explicit feedback is given through an audible cheer and colourful streamers floating across the screen.

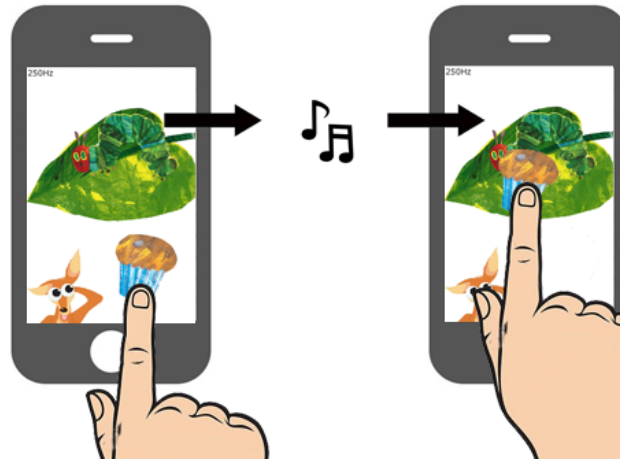


Figure 2- Drag and Drop token in response to tones

LOGIC STRUCTURES

Timers

To perform accurate measurement, two timers have been implemented to determine the amount of time the user has to 'drag-and-drop' a token into a container displayed on screen. The amount of time between each time frame, and the longevity of silence, are determined by pre-programmed variables.

At present these settings are set to default values of a 6000ms response time and a random timer between 5500ms-12500ms for the amount of time to wait between tones played. This random timer resets after each tone is played, resulting in the timed gap between each tone changing throughout the test, potentially preventing users from detecting a recurring pattern in the testing process.

Divide and Conquer Algorithm

The original testing process was based on a linear progression. The process for this was as follows:

- 1.0 Initialise the application and set the frequency to 250Hz and dBHL to -5.
- 2.0 IF the current frequency is less than or equal to 8kHz, play the frequency.
 - 2.1 IF the player responds to the sound AND the response is correct
 - 2.1.1 Store data, increase frequency, and return;
 - 2.2 IF the player does not respond to the sound OR response is incorrect
 - 2.2.1 IF 80dBHL has not been reached
 - 2.2.1.1 Increment dBHL and return;
 - 2.2.2 Else
 - 2.2.2.1 Store data, increase frequency, and return;
- 3.0 ELSE Display the audiogram of results

While this may be a thorough means of testing the participant, there were concerns the testing time may have an impact on the accuracy of test results, with the potential for inaccuracy if the user became uninterested or disengaged. In order to reduce the testing time, and potentially increase accuracy, a different strategy was devised.

Typically, an audiologist will not play all tones procedurally to a participant. The audiologist may commence by playing a middle volume tone, wait for a response and, using their own judgement, observation and professional experience, determine whether the participant can hear a sound. If necessary, the audiologist will play a louder or quieter sound, or even replay a tone multiple times to confirm whether the participant has heard the tone.

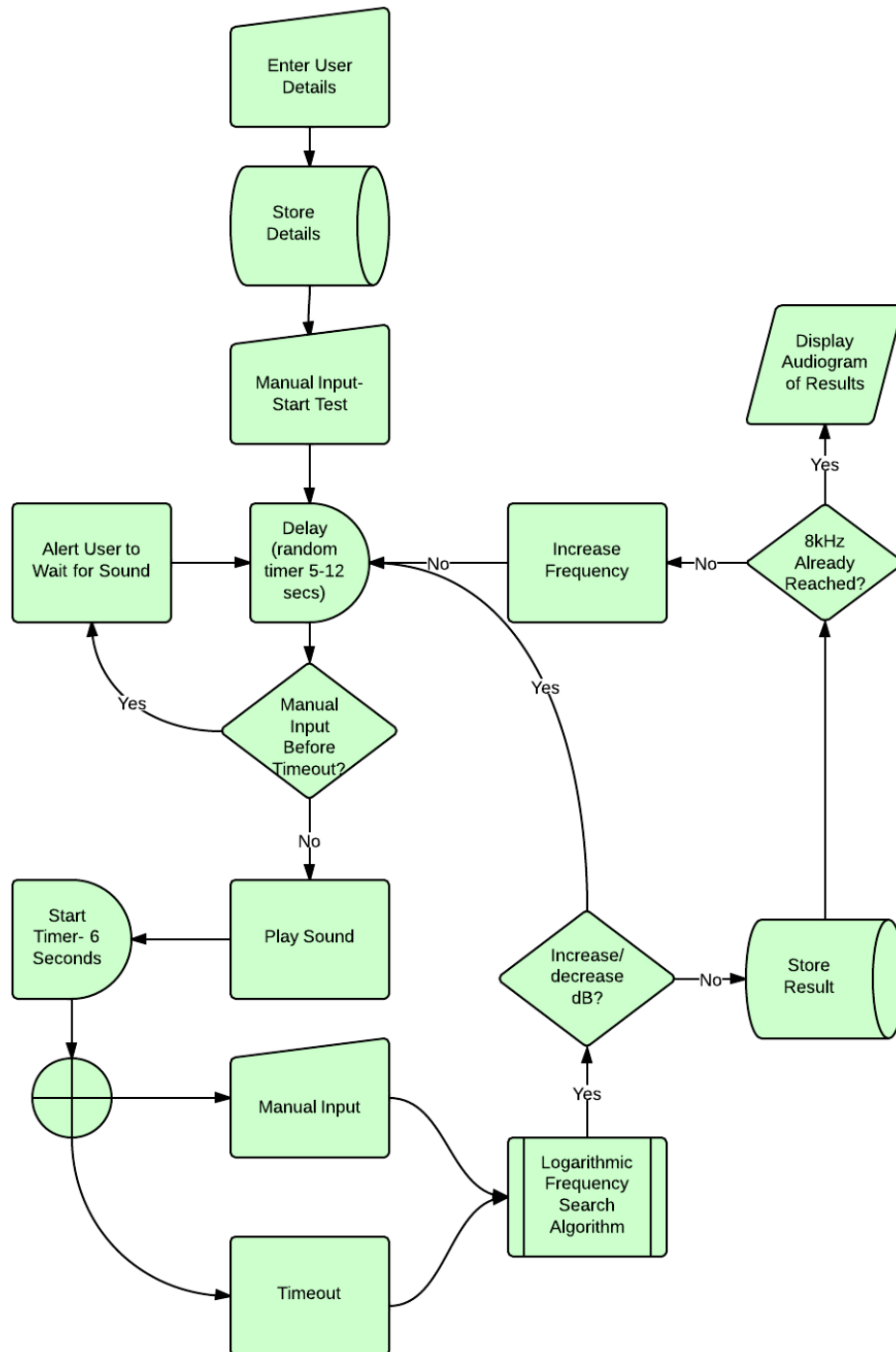


Figure 1- Hearing Test Application Diagram

Unfortunately, without the implementation of a complex intelligent system to learn and adapt for testing of every child, the ability to create an application to emulate an Audiologist is not possible and outside of the project scope. However, using solutions developed for similar applications, a new algorithm was devised based upon commonly used divide-and-conquer methodology, based heavily on the binary search algorithm.

The implementation for the algorithm is as follows:

1. Create a sorted array of tones of the same frequency, in ascending volume order.
2. Get the middle element of the array.
3. Test the user. Two cases are possible:
 - 3.1. IF the user responds, than the result is less than the middle element. In this case, go to step 1 for the part of the array before the middle element
 - 3.1.1. IF the current element can no longer be divided, return current element.
 - 3.2. IF the user does not respond, test again. If the user fails to respond a second time, the result is greater than the middle element. In this case, for to step 1 for the part of the array after the middle element.

The time taken to execute the test for each frequency is dependent on the number of tones, but runs logarithmically as it constantly divides the possible number of results. Figure 3 demonstrates the process of hearing assessment using the application.

DEVELOPMENT PROCESS AND LIMITATIONS

An Audiogram is a graphical representation of an individual's hearing test results. This is a standard format, where different icons are used to mark the graph to denote whether the test conducted was ear specific or bilateral. The initial plan for the application development was to utilise an existing image of a professional audiogram, and overlay the graph by changing icon's global x and y coordinates depending on the frequency and final decibel level. Unfortunately, after initial implementation, there were numerous issues encountered with scaling to the resolutions of different devices. As a result, the audiogram required a complete reworking.

The final solution was to re-create the entire audiogram, with graphing, markers and lines drawn in real-time. This method was implemented by utilising ActionScript's draw functions, within Flash Professional. The final approach now ensures the issues previously experienced with scaling have been eliminated.

AUDIOGRAM OUTPUT

A comparison of an audiogram from an Audiologist (Figure 4) and the Audiogram output from the application (Figure 5) are as follows.

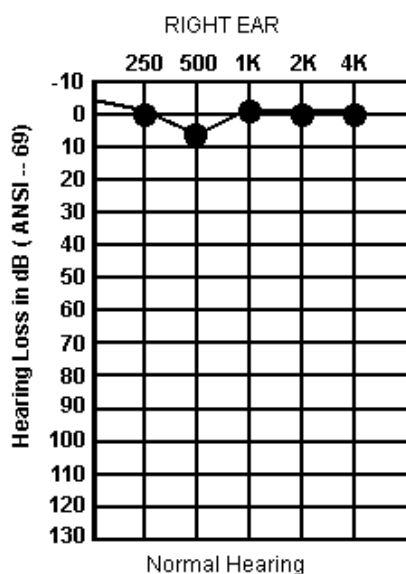


Figure 4-Traditional Audiogram

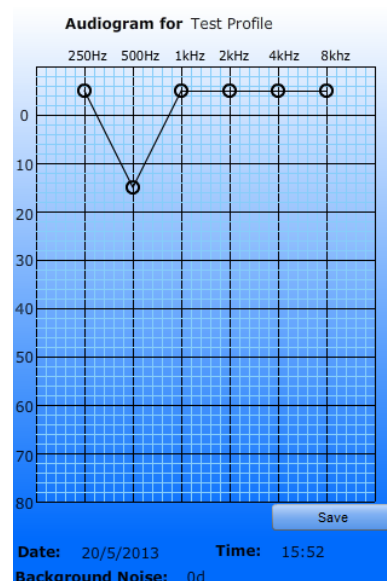


Figure 5- Application Audiogram

BACKGROUND NOISE ANALYSIS

During the initial design process the issue of background noise was a concern. In the pilot study, this will not be a concern, as the research will be conducted in a sound booth, provided by Telethon Speech and Hearing, which is a controlled environment. However, future solutions are currently being investigated for use of the application in uncontrolled environments.

The original approach to the issue was to alter the volume of the tones depending on the background noise, although further investigation found that the processing resources required, and the resulting latency, would most likely hinder the test accuracy.

Instead, a background noise analysis tool has been developed to provide an analysis of environmental background noise. The tool utilises an activity event listener from the microphone class, for detecting microphone activity, which is part of the ActionScript 3.0 library. The intended purpose of the microphone detection activity event is to conserve bandwidth and processing resources. This is achieved by setting a silence level, between 0-100, and disabling the microphone when the background noise drops below a certain level- such as when a user stops speaking ("Working with Sound," 2013).

This feature has since been adapted to sample all sounds by setting the silence level to 0, and sampling the audio recorded by the microphone at regular intervals. Samples are extracted every 500 milliseconds, for 20 seconds, resulting in a total 40 environment audio samples.

Once the audio test has completed the volume level for every sound sampled is extracted, averaged, and the user is alerted to whether their environment is appropriate to perform the test (Figure 6). These results are also included on the audiogram so, if a user's test results differ from previous results, a comparison of the level of background noise can be made to see if it is a contributing factor.

When testing was performed from a desktop computer, results were highly accurate. Upon deployment to iOS devices, the background noise analysis tool's ability to detect noises at a distance was hindered by the built microphone and additional testing is required.

VISUAL STYLE

The visual style of the application is aimed towards children, with bright colours, clear instructions and little use of text to reduce clutter and make the best use of screen real-estate. The simplistic visual elements of the application's posting-box game provide the player with the means to quickly decipher the gameplay options and goals, accommodating those who may not be well accustomed to video games without overloading them with unfamiliar concepts.

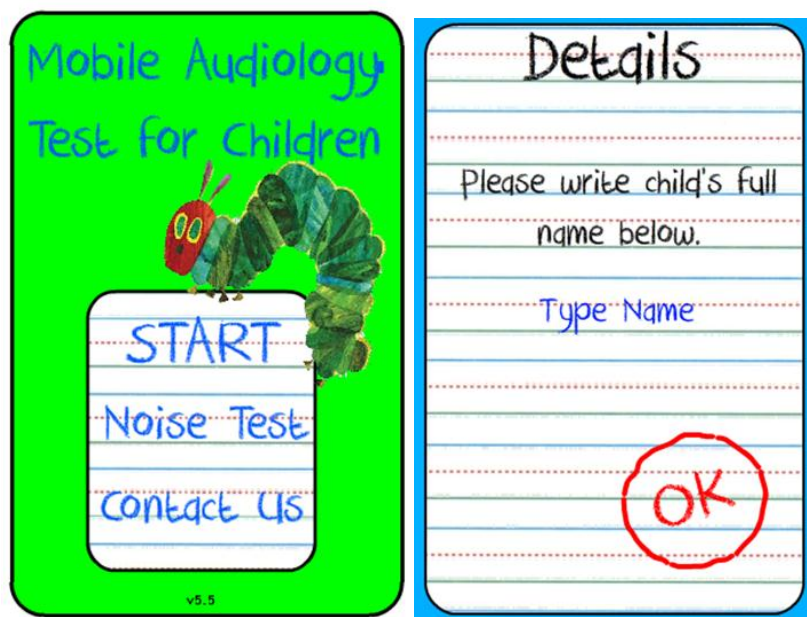


Figure 7. Mobile application user interface

The application utilises images from the children's story book *The Very Hungry Caterpillar* by Eric Carle (Figure 7). Other components such as animations and helper avatar, in the form of a talking kangaroo, provide prompts and feedback to the user.

GAMEPLAY MODULE DESCRIPTORS

The following table (table 1) outlines the gameplay module descriptors for the application, including the five core principles of gameplay design, the primary objective and how the principles are implemented and executed.

Module	Objective	Content/ Media	Interaction	Result
1. Rules	Define basic game play, where player must respond to sounds played by dropping tokens into a container within a specific time frame.	'Helper' avatar described the rules to the player on commencement of the level (future implementation).	The player drags-and-drops token into container if sound is heard.	Player's response to sounds will provide an accurate measurement of hearing.
2. Scoring system	Gives explicit feedback on player's actions, and results of test are collated and displayed in final audiogram.	Token will snap-back to point of origin if no sound has been played. Avatar will provide feedback (future implementation).	The dB (loudness) of the current frequency will increase/decrease according to the LFSA.	Player can see the result of their actions, encouraging listening.
3. Interface	Provide simplistic options and interface for player, no matter their comprehension of technology or serious games.	Colourful and animated imagery and interface. Animations for helper avatar, and during successful drop of token into container (future implementation)	Players touch and swipe their way around the interface.	Reduces chance of player becoming overwhelmed by an overly complex interface.
4. Goals/ Levels	Ensure player is aware of progress through visual indicators.	Structure of the game is in levels of audio frequency to define progress. Different tokens used to denote level increase.	The player selecting the token and dragging into the container, when hearing a sound.	The player is made aware of their progress through visual indicators. Completion of the test shows an audiogram.
5. Feedback	Guide players along the hearing test, giving them explicit feedback through animations and snap-back to prevent using tokens when no sound is playing.	Animations provide feedback based on player's actions.	Player's actions trigger explicit feedback.	Player develops an understanding of game mechanics.

Table 1- Gameplay Module Descriptors

FUTURE DEVELOPMENT AND DIRECTION

The application has been developed with a number of adjustable variables, which may need to be altered during the testing process to ensure accuracy of the test results. These variables include:

- Number and order of tones and frequencies played.
- The number of seconds to wait between tones. This is a random number, decided before each tone is played.
- The number of seconds a user has to respond after a tone has been heard. By default this is set to 6 seconds.

- Implementation of different visuals and auditory presentation of the application content, to cater for multiple demographics.

CONCLUSION

The issue of hearing loss, and causative ear infection, has been a persistent issue in Australian health for many years. A proposed solution is a self-administered audiology testing application for iOS smart phone and tablet devices, emulating an audiogram conducted in an audiometric sound booth. While solutions similar to this application exist, there has been no research conducted into the accuracy and feasibility of a self-administered audiology testing application using mobile devices for children.

As a relatively new field, the results of the application will help to lay the foundation for future research, and potentially reduce the number of children with an undiagnosed hearing loss, through early intervention, improving their quality of life. Additionally, the application has the potential to reduce the strain on current telehealth services, and complement similar solutions to achieve a holistic approach for hearing screening and diagnosis of ear infection.

The application prototype is projected for testing in late 2013, at the Telethon Speech and Hearing institute, Perth, Western Australia. The results are expected to deliver crucial quantitative data to determine the accuracy of the application, as well as qualitative data for feedback from medical professionals, and general consumer response.

REFERENCES

- Adobe Help and Support*. (2013). Retrieved May 20, 2013, from <http://helpx.adobe.com/support.html>
- Bensink, M. E., Brown, C., Elliott, G., Perry, C., Scuffham, P., Smith, A. C., & Stewart, C. (2010). The feasibility of a community-based mobile telehealth screening service for Aboriginal and Torres Strait Islander children in Australia. *Telemedicine and e-Health*, 16, 950+.
- Develop. (n.d.). *iOS Developer Program*. Retrieved May 20, 2013, from <https://developer.apple.com/programs/ios/develop.html>
- iOS Dev Center*. (2013). Retrieved 20th March, 2013, from <https://developer.apple.com/devcenter/ios/index.action>
- iOS Developer Program*. (2013). Retrieved May 20, 2013, from <https://developer.apple.com/programs/ios/>
- Korschun, H. (2012). *Diagnose ear infections at home with iPhone*. Retrieved from http://news.emory.edu/stories/2012/09/remotoscope_for_ear_infections/campus.html
- Leach, A. J. (1999). Otitis media in Australian Aboriginal children: an overview. *International Journal of Pediatric Otorhinolaryngology*, 49, Supplement 1(0), S173-S178. doi: [http://dx.doi.org/10.1016/S0165-5876\(99\)00156-1](http://dx.doi.org/10.1016/S0165-5876(99)00156-1)
- O'Keefe, S. (2012). A hearing loss problem has knock-on effects; In one remote community, more than 90 per cent of children have some form of hearing loss which could be rectified, and it has an impact in the classroom, *ABC Premium News*. Retrieved from <http://ezproxy.ecu.edu.au/login?url=http://search.proquest.com/docview/1031098247?accountid=10675>
- Porting Android to Devices*. (n.d.). Retrieved May 19, 2013, from <http://source.android.com/source/index.html>
- Pigeon, I. S. (2013). *Online Hearing Loss Test and Audiogram Printout*. Retrieved April 5, 2013, from www.myhearingtest.net
- Stevens, Clair. (2013). *The Growing Child : Laying the foundations of active learning and physical health*. Retrieved from <http://ECU.ebilib.com.au/patron/FullRecord.aspx?p=1170331>
- Timms, L., Grauaug, S., & Williams, C. (2012). Middle Ear Disease and Hearing Loss in School-Aged Indigenous Western Australian Children. *Asia Pacific Journal of Speech, Language and Hearing*, 15(4), 277-290. doi: 10.1179/136132812804731794
- Weber, D. (2012). *WA Indigenous children's hearing compared to third world*. Retrieved Jun 22, 2012, from Australian Broadcasting Corporation

<http://ezproxy.ecu.edu.au/login?url=http://search.proquest.com.ezproxy.ecu.edu.au/docview/1021379773?accountid=10675>

What is iOS. (2013). Retrieved May 20, 2013, from <http://www.apple.com/au/ios/what-is/>

Windows Phone- Phones. (n.d.). Retrieved May 20, 2013, from <http://www.windowsphone.com/en-us/phones>

Working with Sound. (2013). *ActionScript 3.0 Developer's Guide*. Retrieved May 20, 2013, from http://help.adobe.com/en_US/as3/dev/WS5b3ccc516d4fbf351e63e3d118a9b90204-7d1d.html